UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
12/787,500	05/26/2010	Kazuaki Ishizaki	JP920080314US1	4722

29154 7590 01/31/2017

FREDERICK W. GIBB, III
GIBB & RILEY, LLC
844 West Street
SUITE 200
ANNAPOLIS, MD 21401

EXAMINER	
BOURZIK, BRAHIM	

ART UNIT	PAPER NUMBER
2191	

NOTIFICATION DATE	DELIVERY MODE
01/31/2017	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

support@gibbiplaw.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte KAZUAKI ISHIZAKI,
KIYOKUNI KAWACHIYA, and KAZUNORI OGATA

Appeal 2015-003825
Application 12/787,500
Technology Center 2100

Before JOSEPH P. LENTIVECH, JOHN R. KENNY, and
AARON W. MOORE, *Administrative Patent Judges*.

MOORE, *Administrative Patent Judge*.

DECISION ON APPEAL

STATEMENT OF THE CASE

Appellants[1] appeal under 35 U.S.C. § 134(a) from a Final Rejection of claims 1–20, which are all of the pending claims. We have jurisdiction under 35 U.S.C. § 6(b).

We reverse.

THE INVENTION

The application is directed to "[a] method, computer system and computer program for optimizing the processing of a character string during execution of the program by using characteristic information that indicates a characteristic of the character string and is associated with the character string." (Abstract.) Claim 1, reproduced below, is representative:

> 1. A method of optimizing processing of a character string during execution of a program, the method comprising:
>
> performing, by a computer system, a first operation for a first character string in the program so as to obtain a second character string comprising multiple characters;
>
> determining, by the computer system based on a characteristic of the first character string and on the first operation for the first character string, characteristic information for the second character string, the determining of the characteristic information for the second character string comprising:
>
> executing a sequential processing of the first character string in the first operation; and
>
> checking characters to be sequentially processed simultaneously with the sequential processing thereby determining the characteristic information of the second character string;

---

[1] Appellants identify International Business Machines Corporation as the real party in interest. (*See* App. Br. 3.)

associating, by the computer system, the characteristic information with the second character string;

storing, by the computer system, the characteristic information, which is associated with the second character string, in memory such that the characteristic information, as stored, is expressed as a bit string comprising multiple bits with each bit in the bit string corresponding to a different information piece that characterizes the second character string as a whole and not individual characters within the second character string; and

performing, by the computer system, a second operation for the second character string in the program, the performing of the second operation being in accordance with the character information associated with the second character string, thereby optimizing the performing of the second operation.

THE REFERENCES

The prior art relied upon by the Examiner in rejecting the claims on appeal is:

Lee et al.	US 2001/0054031 A1	Dec. 20, 2001
Arnold et al.	US 6,523,168 B1	Feb. 18, 2003
Koseki et al.	US 2005/0231397 A1	Oct. 20, 2005
Rabetge et al.	US 2008/0288549 A1	Nov. 20, 2008

THE REJECTIONS

1. Claims 1, 3, 5, 13, and 15–20 stand rejected under 35 U.S.C. § 103(a) as unpatentable over Koseki, Arnold, and Rabetge. (*See* Final Act. 10–21.)

2. Claims 2, 4, 6–12, and 14 stand rejected under 35 U.S.C. § 103(a) as unpatentable over Koseki, Arnold, Rabetge, and Lee. (*See* Final Act. 21–34.)

ANALYSIS

The Specification provides an example of how the claimed system may be used to optimize the processing of character strings during operation of a program:

> In a case where the characteristic information indicates [that the string has no upper case characters, or no lower case characters], the processing to convert a character string into a lower case character or processing to convert a character string into an upper case character may be omitted during execution of String.tolowerCase() method or String.toUpperCase() method.

(Spec. 20.) In other words, for a string like "OPEN," the first operation may determine that the string has, for example, no lowercase characters, that "characteristic" of not having any lowercase letters may be stored, and a programmatic step that calls for a conversion of all lowercase letters to all uppercase characters may be skipped for that particular string based on the characteristic.

Claims 1–17, 19, and 20

The Examiner finds that Koseki teaches all of the limitations of claim 1, except for those regarding storage of the characteristic information as a bit string, for which Rabetge is cited, and those regarding the second operation, for which Arnold is cited. (*See* Final Act. 10–17.) Koseki describes a compiler "for optimization of conversion of a character coding system for a character stored in a string variable" and, in particular, "for optimization of a conversion instruction for conversion from UTF8, which is a character coding system for characters in XML documents, to UTF16, which is a character coding system used in a case where a string of characters is manipulated by" a Java program. (Koseki ¶ 23.) Rabetge teaches the use of a bit string to represent whether each character in a string is upper case or

lower case. (Rabetge ¶ 11.) Arnold teaches that Java includes a string concatenation operation. (Arnold 2:30–37.)

Appellants argue that "while Rabetge does disclose the use of a bit string to store information, the bits in the Rabetge bit string do not represent the same information as the bits in the claimed bit string and, particularly, do not characterize all characters in a character string." (App. Br. 17.) We agree. Rabetge describes how "[i]n web service development environments, pathnames must be created which are case-insensitive and which do not allow for special characters" such that "pathnames need to be created in the development environment to ensure that different web services can be uniquely identified." (Rabetge ¶ 2.) Rabetge's solution for a given identifier (e.g., a pathname) is to generate a bit string that "identifies whether each letter in the [identifier] is upper case or lower case." (*Id.* ¶ 3.) Then, "upper case letters in the [identifier] are converted to lower case" and "[t]he converted [identifier] is concatenated with the bit string to generate a case-insensitive [identifier]." (*Id.*) In other words, the string is converted to all lowercase, but the bit string retains a record of which individual characters had been upper case before the conversion. Thus, Rabetge's bit string identifies a case for each individual character, and does not "characterize[] the [] character string as a whole *and not individual characters within the [] character string*," as claimed. The Examiner's interpretation of the claim as covering a string that characterizes individual characters (Ans. 38) cannot be squared with this claim language.

Because we conclude the combination does not teach or suggest all of the limitations of claim 1, we decline to sustain (a) the Section 103(a) rejection of claim 1, (b) the Section 103(a) rejections of independent claims

18 and 20, which include the same limitation regarding the bit string, and (c) the Section 103(a) rejections of dependent claims 2–17, all of which also include that limitation. We do not reach Appellants' other arguments regarding these claims.

Claim 19

Claim 19 recites that the "bit string compris[es] multiple bits with each bit in the bit string corresponding to a different information piece that characterizes the second character string," but omits "and not individual characters within the second character string" as in the other independent claims. Claim 19 does, however, include a similar limitation in the "determining" portion of the claim: "the characteristic information comprising at least one information piece about the second character string *as a whole and not individual characters within the second character string*." Because Koseki also employs a character-by-character approach,[2] we agree with Appellants (*see* App. Br. 34), for essentially the same reasons as described above regarding the bit string limitation of the other claims, that Koseki does not teach or suggest this limitation. We, therefore, do not sustain the rejection of claim 19. We do not reach Appellants' other prior art arguments regarding these claims.

As the Examiner observes (Ans. 79), the objection to claim 19 is not appealable and should be addressed either with the Examiner or by petition should prosecution continue.

---

[2] *See, e.g.,* Koseki ¶¶ 23–25, 34–35 ("[T]he constructor generates coding system information 210 indicating that *a character* in the string variable is UTF8 or UTF16.").

DECISION

The rejections of claims 1–20 are reversed.

REVERSED